Excise movement and control system (EMCS)

Beskrivelse af struktur og design af de funktionelle beskeder

Versionsstyring

Dato	Version	Udarbejdet af	Beskrivelse af ændring
05-04-2010	1.0	SKAT	Første version.
27-09-2010	1.1	SKAT	Mindre rettelser.
Februar 2017	1.2	SKAT	Opdatering af link til Unicode standard release 6.1.0

1. Læsevejledning

Dette dokument indeholder uddrag fra DDNEA, EU-kommissionens designdokumentation (gengivet på engelsk), som beskriver hvordan de funktionelle beskeder er bygget op, hvilke design valg der er fortaget omkring indholdet af beskederne, samt generel information om XMLdokumenter.

2. EMCS Technical Message Structure

2.1. *Introduction*

In this section the basic elements of the EMCS Technical Message Structure are described. The elements are devided into 3 items:

- Data Items;

- Data Groups;

- Codelists (sets of discrete values).

2.2. *Data dictionary*

2.2.1. Data Items

Every Data Item is identified by a unique name. Note that every name will in principle contain some lowercase characters, except for the following:

- Trader ID;

- NAD LNG.

Every Data Item has an associated type (which can be numeric, alphanumeric or decimal) and in some cases a Data Item can have only discrete values, in which case the Data Item is said to have an associated codelist.

It shall be noted that there are two categories of free text fields within EMCS:

- Fields with an associated language code (LNG field). This LNG field may contain the code of the language in which the text was originally written;

- Fields without such language code.

2.2.2. Data Groups

Every Data Group consists of a number of Data Items in a particular order. Every message is composed of a certain number of Data Groups in a particular hierarchy. Every Data Group is identified by a name. To be noted is that group names are not unique. It may thus very well happen that the same group name is found in different messages. Moreover, Data Groups with the same name do not always include the same Data Items. Hence, when a Data Group is used in more than one place including different Data Items each time, then this Data Group should be assigned to all of these Data Items even if not all of the Data Items are used in every instance.

To be noted is that some Data Groups may not always have the same Data Item sequence in different messages.

2.2.3. Codelists

A codelist is a set of discrete values, with an associated meaning (*Kodelisten kan findes på hjemmesiden*).

A name and a number identify codelists. Codelists are maintained by the SEED. The Central Project Team, supported by the Legal and Procedural team, will maintain the business codelists on the central reference site. The MSAs can then download the new codelists from this reference site. There are a number of technical codelists for which the values are predefined and fixed. These values are not maintained within the common reference data. These codelists are marked as technical codelists.

2.3. *Technical message structure (TMS)*

The structure and format of the different Information Exchanges (IE) are included in the corresponding appendix (*Dette appendiks er ikke vedlagt, men er erstattet af Excel dokumentet "Oversigt over beskedstruktur vX.X" som indeholder ét ark per IE besked*). These appendices contain a message format description for every Information Exchange that is part of the particular system.

The technical message description is supplied in two parts.

The first part is the overall message description. This description contains the overall layout of the messages. It defines the different Data Groups that are part of the message, the sequence of the groups, the level of hierarchy of the Data Groups, the optionality of the Data Group, the possible repeat count, and associated rules and conditions. Concerning the optionality, it should be noted that the following rules apply:

- If a Data Group is always required, it is marked as "R";

- If there exist one or more conditions related to the presence of the Data Group, it is marked as 'D'. When a condition indicates that a dependent data group "does not apply" in a specified case then the specific data group must not be present in the message structure;

- If a Data Group is not always required and there are no conditions related to its presence, it is marked as 'O', meaning that the Data Group may either be present in the message structure or not. However, if information is available it is recommended to be included in the message despite the fact that this Data Group is characterised as Optional.

In order to go down one level in the hierarchy, the Data Group at the higher level in the hierarchy needs to be present. The second part of the TMS contains the description of the different Data Items. This description includes the sequence of the data elements in the group, the optionality, and the associated rules and conditions.

Concerning the optionality of the Data Items, the following rules apply:

- If a Data Item is always required, it is marked as "R";
- If there exist one or more conditions related to the presence of the Data Item, it is marked as 'D'. When a condition indicates that a dependent data item "does not apply" in a specified case, then the specific data item must not be present in the message structure. It shall be noted that a Data Item set to "NULL" (empty valued) is still present in the message structure. Hence, when a Data Item is marked as "does not apply" in a specific case, the generated messages

shall not include it with a "NULL" value. Subsequently, when all Data Items of a Data Group are set to "NULL", the Data Group is still present in the message structure. Hence, when a Data Group is marked as "does not apply" in a specific case, it shall not be present in the generated messages with all of its Data Items set to "NULL".

- If there are no conditions related to the presence of a particular Data Item, it is marked as 'O', meaning that the Data Group may either be present in the message structure or not. However, if information is available it is recommended to be included in the message despite the fact that this Data Group is characterised as optional.

The rules and conditions are marked as RXXX and CYYY. When data is derived from another message, the rules and conditions are implicitly carried forward.

The message description part of this document consists of message hierarchies and correlation tables in order to map the Information Exchanges to those hierarchies (XML).

It should be noted that the Data Items are characterised as "Dependent" when *Conditions, Rules* or *Technical Rules* are applied to them. The letter "D" illustrates precisely the dependency on different entities. An indicative example is the following:

The Data Item *(DELIVERY PLACE) TRADER. Trader ID* is dependent (D) since both the C074 and R045 are applied to it. According to C074, the presentation of this Data Item (if it is "Required" or not) is dependent on the DESTINATION TYPE CODE value. Moreover, according to R045, the content of this field (if it will be the excise number, the temporary authorisation, VAT number, etc.) is dependent on the DESTINATION TYPE CODE.

2.4. *Common Message Header*

The "Message Header" is common for all messages and consists of the following Data Items:

- Message sender and Message recipient: The Data Items "Message sender" and "Message recipient" are both "Required" and should contain, respectively, the address of the sender and recipient, defined as:

    <Application Name>.<Country ISO Code>,
    where Application Name can have the following value: NDEA (Nationally Developed Excise Application);

- Date of preparation: This is a "Required" Data Item used for the date that the Information Exchange was put into XML representation (generation of the XML message);

- Time of preparation: This is a "Required" Data Item used for the exact time that the Information Exchange was put into XML representation (generation of the XML message);

- Message identifier: This is a "Required" Data Item generated by the sending application to uniquely identify the information exchange;

- Correlation identifier: This is a "Dependent" Data Item, since it is only Required for correlating the response and refusal messages. It does not apply for requests and one way messages.

3. Design Principles

Due to the fact that Information Exchanges are used to update data of Excise operations held by different applications, not all data is uniquely identifiable, and therefore, the following rules are applied for the updating of operation data:

☐ Key fields: The ARC is a key to uniquely identify EMCS operations. It is unique and it refers to EMCS movements. Each e-AAD body is uniquely identified by its "Body Record Unique Reference" number within an ARC;

3.1. *Data Item conventions*

Every Data Item within a TMS can be numeric, alphanumeric, text field, dateTime, date, or time. A number of rules and conventions have been defined for the possible data formats when present in the Common Domain. These rules are the same for data exchanged in XML format.

3.1.1. Numerical Fields

Concerning numerical fields, it should be noted that these contain either a cardinal value (positive integer value) or a decimal value.

The decimal separator is the decimal point ".". No other symbols are permitted as decimal separator. Triad separators, such as a comma, shall not be used.

Signs, whether positive or negative, shall not be used (all values are always intrinsically positive). For decimal values, the decimal notation (with the decimal point) should only be used when there is a reason to indicate precision. E.g., for a mass value:

- 89 kg, with a precision of 1 kg.

- 89.2 kg, with a precision of 0.1 kg.

- 89.20 kg, with a precision of 0.01 kg.

For numerical values, leading zeroes shall not be used. Trailing zeroes should only be used to indicate precision.

If the decimal point is present, at least one digit shall be present before the decimal point.

If the decimal point is present, at least one digit shall be present after the decimal point.

Examples for a n..11,3 type.

12345678.123	(Valid).
123456789012.123	(Invalid - too many digits before decimal point and hence too many digits in total).
12345678.1234	(Invalid - too many digits after decimal point and hence too many digits in total).
0123	(Invalid - leading zero not permitted).
+123	(Invalid - plus sign not allowed).
-123	(Invalid - minus sign not allowed).
1,234	(Invalid - triad separator not allowed).
.3	(Invalid - no digit before decimal point).
12345.	(Invalid - no digit after decimal point).

0.3 (Valid).

1.3E1 (Invalid - only digits and decimal point allowed).

12345678901 (Valid - n..11,3 can have maximally 11 digits of which maximally 3 after decimal points).

It is to be noted that the rules above also apply to numerical values within codelists. Values in such a list should always be stored without leading zeroes (in order to avoid problems of comparing e.g. a value of 60 against a value of 060). If the leading zeroes are omitted, a numerical comparison should always work out fine.

It should be noticed that there are no codelists with decimal values.

3.1.2. Date/Time Fields

The specification of Date and/or Time fields used in TMS is as per W3C XML Schema specification XML Schema 1.0 http://www.w3.org/TR/xmlschema-0/ except that:

- for all times in DateTime and Time fields time zone must be omitted with the local time always implied as being the Coordinated Universal Time (UTC, sometimes called "Greenwich Mean Time");

- all years in DateTime and Date fields are in the Common Era (i.e. AD), hence the negative sign is not permitted.

Although the reader should refer to the W3C XML Schema specification (XML Schema 1.0 http://www.w3.org/TR/xmlschema-0/), the following table (Table 1) indicates the format for each type and their corresponding regular expression.

Type	Regular Expression
Date	yyyy '-' MM '-' dd \d{4}-\d{2}-\d{2}
Time	hh ':' mm ':' ss ('.' s+)? \d{2}:\d{2}:\d{2}(\.\d+)?
Date/Time	yyyy '-' MM '-' dd 'T' hh ':' mm ':' ss ('.' s+)? \d{4}-\d{2}\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?

Table 1: Date/Time fields format and their corresponding regular expressions

Where:
- yyyy is a four-or-more digit;

- the remaining '-'s are separators between parts of the date portion;

- MM is a two-digit numeral that represents the month;

- dd is a two-digit numeral that represents the day;

- 'T' is a separator indicating that time-of-day follows;
- hh is a two-digit numeral that represents the hour; '24' is permitted if the minutes and seconds represented are zero, and the dateTime value so represented is the first instant of the following day (the hour property of a dateTime object in the ·value space· cannot have a value greater than 23);

- ':' is a separator between parts of the time-of-day portion;

- mm is a two-digit numeral that represents the minute;

- ss is a two-integer-digit numeral that represents the whole seconds;

- '.' s+ (if present) represents the fractional seconds.

3.2. *Character set usage*

Messages exchanged in XML format shall use the UTF-8 encoding of UNICODE both for language sensitive fields and for non-language sensitive fields.

3.3. *Language Indicator for Language-sensitive text fields*

The associated LNG indicator, if present, denotes the language in which the original text was written. From this, the character set used can be derived. All relevant information can be found in the Unicode standard Release 6.1.0 **http://www.unicode.org/versions/Unicode6.1.0/.**

4. XML formatting

This section defines how the messages need to be formatted in XML format. In particular, it specifies the XML conventions for EMCS as well as the Character Sets that shall be supported by NDEAs.

4.1. *XML Schema*

The XML (Extensible Markup Language) is a subset of SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, its goal is to enable generic SGML to be served, received and processed on the web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML. The complete specific of XML 1.0 can be found at http://www.w3.org/TR/1998/REC-xml-19980210[1]. The XML schema definition language (XML Schema 1.0 http://www.w3.org/TR/xmlschema-0/) is a model for describing the structure of information. In particular, the XSD defines a type of XML document in terms of constraints upon what elements and attributes may appear their relationship to each other and what types of data may be in them. The schema language, which is introduced in XML 1.0 and uses namespaces, considerably extends the capabilities found in XML 1.0 document type definitions (DTDs).

The Design Document for National Excise Applications imposes that XML formatted messages are valid, UTF-8 encoded [Unicode standard Release 6.1.0 http://www.unicode.org/versions/Unicode6.1.0/], XML 1.0 [XML standard 1.0 Release 3 http://www.w3.org/TR/REC-xml] documents that follow the rules defined in the XML Schema [XML Schema 1.0 http://www.w3.org/TR/xmlschema-0/].

---

[1] *An interesting annotated version of the specification is available at http://www.xml.com/axml/testaxml.htm.*

4.1.1. XML namespaces

The XML format specification makes use of the name space concept. XML namespaces provide a simple method for qualifying element and attribute names used in XML documents. The use of different namespaces visually identifies the source schema for a XML element in an instance and enables the logical separation of elements that represent the XML Entity Model, XML Entity Actions and XML message definition.

The significant namespaces are:

- "ea" to qualify elements and attributes related to the XML Entity Actions;

- "msg" to qualify elements and attributes related to the definition of a XML message;

- "exc" to qualify excise domain specific elements and attributes;

- "lsd" to qualify language specific elements and attributes;

4.2. *Character set support*

XML document and schema always use one encoding, which is specified in the prologue of the document and schema. Currently this encoding will be UTF-8. The EMCS system (or NDEA application) will support the UTF-8 character set encoding for all message exchanges.